System for Geometrically Accurate Compression and Decompression

By:

Stanley Osher
and
Hong-Kai Zhao

Atty. Docket No.: 03.002

Law Offices of Matt Rainey
1241 Grizzly Peak Blvd.
Berkeley, California 94708-2127
Tel: (510) 540-5580

Background of the Invention

This application claims the benefit of U.S. Provisional Application No. 60/257,210, of Osher et al., filed December 20, 2000, entitled "Geometrically Accurate Compression and Decompression", which is incorporated herein by reference.

5   The present invention relates to a system for compressing and reconstructing (decompressing) signals, including signals representing physical data such as terrain. Such signals may be computer-constructed data, empirically derived images or signals, or in general any information or data representing actual or simulated phenomena.

After compression of data representing a terrain, digital terrain elevation (or

10   images, indeed any graph) data ("DTED"), a common problem is accurate reconstruction of terrain features, such as the slope. Given digitized data representing terrain elevation, or indeed any graph in two or more dimensions at a data point (x,y) (or the multiple dimension extension at a data point $(x_1,\ldots,x_n)$), it is highly useful to be able to compress and decompress the data in such a way that terrain -- or, more generally, graph or image

15   features – are accurately reconstructed.

Conventional systems in current use compress the data, store or transmit it as needed, and at a later point reconstruct the data from the compressed-data files. This can lead to large errors in derived quantities obtained from the reconstructed data, in particular when the gradient of the reconstructed data is taken.

20   The gradient of the original data is often of considerable interest. In the case of DTED, it may be important for aircraft to be aware of the precise terrain slopes, and the errors introduced by determining gradients from reconstructed data may be too great to be of practical use for many applications.

Especially for lossy compression techniques, a reconstructed image generally does

25   not have the same values for the norm of the gradient that the original image had. In fact, errors in the gradient are often quite large for higher compression ratios. These errors are

quite significant since the accuracy of the terrain slope is crucial in many areas, e.g. landing of aircraft in navigation exercises.

Accordingly, a system is needed that can compress and reconstruct multidimensional data, such as terrain data or other signals, with an increased accuracy in

5   the reconstructed information, and in particular in the gradients that are determined from the reconstructed data, and the norms of those gradients.

Summary of the Invention

An apparatus and method according to one embodiment of the present invention

10   are implemented in a processor-based system. A method according to the invention provides an accurately compressed and reconstructed signal, and in particular accurately reconstructed gradients of the original signal, by applying compression procedures to the gradient of the original signal. An alternative embodiment further involves the process of generating a second gradient of the original signal, and compressing that second gradient,

15   which upon decompression provides an accurate reconstruction of the second gradient of the original signal. The methods of the invention are suitable for compressing, storing, transmitting and reconstructing both simulated and empirical data representing terrain and other physical or hypothetical signals or surfaces, in one or multiple dimensions.

20   ## Brief Description of the Drawings

Figure 1 is a block showing a processor-based system suitable for an implementation of the present invention.

Figure 2 is a representation of a signal that can be compressed and reconstructed according to the present invention.

25   Figure 3 is a representation of a gradient of the signal of Figure 2.

Figure 4 is a representation of another signal that can be compressed and reconstructed according to the present invention.

Figure 5 is a representation of a gradient of the signal of Figure 4.

Figure 6 is a representation of a gradient of the signal of Figure 5.

5   Figures 7-8 are flow charts representing methods implementing aspects of the present invention.

Description of the Preferred Embodiments

Figure 1 illustrates a processor-based system 100, such as a workstation or a

10   server, in connection with the present invention may be implemented. The system 100 is coupled to a display 110 and one or more user interface devices 120 (such as a mouse, keyboard, track ball, etc.), and operates under control of at least one microprocessor 130 (though it may be a multiprocessor system).

The processor 130 is connected to a local device control circuitry 140, which

15   includes circuitry that controls data and command exchanges with local devices such as the display 110 via an accelerated graphics processor (AGP) 150 and memory 160, which may in a typical system will include multiple DIMMs (dual in-line memory modules) or other suitable memory modules or media.

The local device control circuitry 140 is connected to a peripheral device control

20   circuit 170 via a bus such as PCI bus 180. The system 100 additionally is connected to local internal and/or external storage 190.

Input/output (I/O) channels 200 are connected to or in communication with the system 100, and may include conventional network connections, wireless communications devices, and/or other conventional apparatus for exchanging data with the

25   system 100. In addition, transmitters, receivers or in general transceivers 210 may be

connected to or in communication with the system 100, which are suitable for remote or isolated operations include weather stations or other telemetry stations, on aircraft, etc.

The compression and decompression operations of the present invention are in one embodiment carried out under the control of program modules stored in the memory

5   160 and executing on the processor 130. These operations may be implemented in and/or executed by software, hardware (e.g. in configured field gate arrays, custom logic, etc.), firmware or some combination thereof. In this application, the term "logic" will be used to refer to any such appropriate combination of software, hardware, firmware or other manner of implementing the invention.

10 The image, terrain, or other data relating to the invention will typically take the form of data files stored in memory or in some storage medium, which can be read, modified, stored, output, displayed and printed by the computer system, either under automatic (or program) control or under the direction of a user.

15 <u>Compression and Reconstruction of Gradient Data</u>

The system of the invention is applicable to many types of compression procedures presently in use, with improved accuracy of the reconstructed signals, and in particular of their gradients (or slopes) and corresponding norms of those gradients. An example of one appropriate compression procedure is a histogram equalization procedure,

20 as described, for instance, in Sapiro, G. and Caselles, V., "Histogram Modification via Differential Equations", *Journal of Differential Equations*, Vol. 135, No. 2, pp. 238-268 (1997), which is incorporated herein by reference. Generally, appropriate compression procedures include those that are exact for constant signals, but are lossy in the general case (such as JPEG or histogram equalization).

25 Thus, the inventive methods may be applied to signals processed according to any compression procedures. In addition, the inventive methods may be applied to signals

representing data of many dimensions. However, for the sake of the following discussion a two-dimensional signal will be taken as an example.

The normal vector to the signal or graph representing a surface or terrain is given by:

5

$$\frac{(-u_x, -u_y, 1)}{\sqrt{(1 + u_x^2 + u_y^2}}.$$

It follows that the steepness (or slope) of the terrain surface is determined by the gradient of that signal, namely:

$$|\nabla u| = \sqrt{u_x^2 + u_y^2}.$$

10   A method according to the present invention compresses the gradient $|\nabla u|$, rather than the signal u itself. The basic framework of such a method is carry out the following operations:

Operation #1: Compute $|\nabla u|$ either analytically or numerically. (It may be done

15   analytically for simulated data, and numerically for empirical data, or even analytically for empirical data that has been approximated by mathematical representations.)

Operation #2: Use any suitable compression technique to compress and store (or transmit) $|\nabla u|$ as $\overline{|\nabla u|}$.

Operation #3: Recover a reconstructed signal v by solving the equation:

20   **Equation A:** $|\nabla v| = \overline{|\nabla u|}$.

In particular, this can be solved using fast numerical level set based procedures for solving the Eikonal equation $|\nabla v| = \overline{|\nabla u|}$.

Figure 7 is a flow chart according to this series of operations.

An Eikonal equation is one of the form

$$\sqrt{v_x^2 + v_y^2} = C(x,y) \quad \text{for } C(x,y) > 0.$$

This gives a generalized distance to a set. If $C \equiv 1$, then the distance is a real distance. Examples of this may include: (a) distance to the origin of a coordinate system, and (b) distance to a set defined by the equation x=0.

5   *(a) Distance to the origin.*

If $v = \sqrt{x^2 + y^2}$, the distance is to the origin *x=y=0*, with:

$$v_x = \frac{x}{\sqrt{x^2+y^2}}; \ v_y = \frac{y}{\sqrt{x^2+y^2}}; \text{ and}$$

$$v_x^2 + v_y^2 = \frac{x^2+y^2}{x^2+y^2} = 1$$

(b) Distance to a set x=0.

10   If $v = |x|$, the distance represents a distance to the set *x=0*, with:

$$v_x = 1 \ \text{ if } x > 0;$$

$$v_x = -1 \ \text{ if } x < 0; \text{ and}$$

$$v_y = 0;$$

and thus $v_x^2 + v_y^2 = 1$. At least a few values of $v = u$ at a few data points are input,

15   including certain boundary points and points of extrema of *u*.

A system according to the present invention can take advantage of fast methods to compute the unique viscosity solution to this Hamilton-Jacobi nonlinear partial differential equation, i.e. the Eikonal Equation A above. The viscosity solution is used because $\left|\overline{\overline{\nabla u}}\right|$ represents $\dfrac{1}{velocity}$ in the Eikonal Equation. Thus, we can view

20   Equation A above as finding the distance *v* in that variable metric.

Operation #3 above involves the solution of

$$\left|\nabla v\right| = f(x,y),$$

where $f(x,y) = \left|\overline{\overline{\nabla u}}\right|$ is the compressed quantity. As mentioned above, this may be a numerical or an analytical solution.

It has been proven that there exists a unique viscosity solution to this equation, given the values of u at appropriate data points. See Rouy, E. and Tourin, A., *A viscosity* 5 *solutions approach to shape from shading*, SIAM Journal of Numerical Analysis, Vol. 29, No. 3, pp. 867-884 (1992). There are fast Dijkstra-like algorithms and/or sweeping algorithms that are designed for this purpose. On Dijkstra-like algorithms, see: J. N. Tsitsiklis, "Efficient Algorithms for Globally Optimal Trajectories", IEEE Transactions on Automatic Control, Vol. 40, No. 9, September 1995, pp. 1528-1538, which is 10 incorporated herein by reference.

To solve the Eikonal equation in any number of independent variables (x, y, z, ...) on a distance grid with n points and values of u assigned at isolated points, there is a shortest-path type of algorithm given by Tsitsiklis, which runs in optimal time $O(n \log n)$, with n being the number of pixels. Dijkstra approach is a classical algorithm 15 which computes the "taxicab" distance metric, i.e. which solves:

$$\max\left(\left|u_x\right|,\left|u_y\right|\right) = C(x,y)$$

in this optimal time. Tsitsiklis generalized it to the true geodesic distance (as the crow flies). The classical algorithm and its generalization update each grid point once in increasing order of distance, and may use an $O(\log n)$ heapsort search.

20 Fast sweeping algorithms solve the same algebraic expression as the Dijkstra-like algorithms on the grid at each point – however, not in increasing order of distance, but rather in an iterative fashion, updating points as often as needed until convergence within a predetermined tolerance. Fast sweeping methods can involve simplified programming and can be faster, e.g. if $C(x,y) \equiv 1$ and a left-right, up-down procedure is used. See, e.g.,

M. Boue and P. Dupuis, *Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control*, SIAM J. Numer. Analysis, Vol. 36, No. 3, pp. 667—695 (1999).

Input used for these fast solution techniques (which can be referred to as "fast solvers") are the numerical values of $f(x,y)$ and the values of u at grid points where $|\nabla u|$ is less than some very small tolerance (i.e. where u might be an extremum).

This present invention is thus useful in combination with any compression routine.

One might consider the approach of compressing the vector $\nabla u = (u_x, u_y)$ to get $\overline{\nabla u} = (\overline{u_x}, \overline{u_y})$, and then recover u. There are two conditions that must be considered to avoid inaccuracies in the reconstruction of the gradient:

(i) There is a need for compatibility, i.e.

$$(\overline{u_x})_y = (\overline{u_y})_x,$$

at least approximately, and this is generally false for the recovered u, using this simplified procedure.

(ii) Reconstructing u from one compressed derivative, e.g. solving numerically $v_x = \overline{u_x}$ can lead to large errors in the resulting v_y.

Reconstruction of the signal from the decompressed gradient is effectively an integration process, which can be carried out numerically or can be carried out analytically for a function derived from or representing the decompressed signal (within some predetermined level of accuracy). Such integration can be carried out for each level of gradient operation (e.g. for second gradients --see Alternative Method I, below).

Other implementations of the basic method for solving $|\nabla u| = \overline{|\nabla u|}$ can be enhanced in several ways. Figure 8 is a flow chart illustrating an Alternative Method I, as follows.

Alternative Method I

5   (1) Compute numerically $|\nabla u|$.

          (2) Compute numerically $|\nabla|\nabla u||$.

          (3) Compress and store $|\nabla|\nabla u||$, obtaining $\overline{|\nabla|\nabla u||}$.

          (4) Recover v using fast a Eikonal solver <u>twice</u>. First solve for w in

$$|\nabla w| = \overline{|\nabla|\nabla u||}.$$

10   Then solve for v in

$$|\nabla v| = w.$$

Wherever numerical computation or data generation is called for herein, it should be understood that under appropriate circumstances an analytical solution may be generated, and vice-versa. In either case, a predetermined level of accuracy may be used a

15   controlling factor, e.g. for the compression, gradient, integration and/or reconstruction operations.

The extra data storage that would be used for this algorithm is the storage of $v = u$ and $|\nabla v| = |\nabla u|$ at extrema, i.e. those points for which $|\nabla u|$ and $|\nabla|\nabla u||$, respectively,

20   have values below a small predefined tolerance.

This method can be generated recursively to N Eikonal solvers. The extra storage is minimal, but the decompression step would be approximately N times slower.

Alternative Method II

5   (1) Solve $\left|\nabla w\right| = \left|\overline{\overline{\nabla u}}\right|$.

     (2) Compress $\left|\nabla(u - w)\right|$, and obtain $\left|\nabla\overline{\overline{(u - w)}}\right|$.

     (3) Solve $\left|\nabla z\right| = \left|\overline{\overline{\nabla(u - w)}}\right|$.

In steps (1) and (3), use the values of u and $u - w$, respectively, at their approximate extrema, as usual.

10   (4) Then obtain v, the reconstruction of u, via $v = w + z$.

This method adds a correction to the basic method by compressing and reconstructing errors in $\left|\nabla u\right|$.

Recovering Curvature

15   Additionally, other geometric features of the terrain, for example mean curvature of the surface, can be recovered from compressed data in a similar manner. Following is a procedure for recovering curvature.

For a surface defined by

$$z = u(x,y),$$

its mean curvature is:

$$k = \frac{u_{xx}(u_y^2 + 1) - 2u_{xy}u_x u_y + u_{yy}(u_x^2 + 1)}{(1 + u_x^2 + u_y^2)^{3/2}}.$$

5   See S. Osher and J.A. Sethian, "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton--Jacobi Formulations", Osher, S., and Sethian, J.A., Journal of Computational Physics, Vol. 79, pp. 12-49 (1988), which is incorporated herein by reference.

This involves first and second partial derivatives of u. Thus, in order to recover

10   curvature from compressed data, Alternative Method I above can be used, with $N \geq 2$. This allows the recovery of a signal v whose second and first derivatives are accurate approximations to those of u. Then a numerical implementation for the expression defining k above may be used, where v replaces u.

Higher derivatives of curvature -- in fact all geometric features -- can be recovered

15   in an analogous fashion, which will be clear from the foregoing to those skilled in the art. For instance, instead of using the Eikonal equation, one can take any elliptic differential operator of second or higher order, for example the Laplacian

$$\mathcal{L}(v) = u_{xx} + u_{yy},$$

and solve numerically

20

$$\mathcal{L}(v) = \overline{\overline{\mathcal{L}(v)}}$$

for v, where appropriate boundary conditions are imposed. Then geometric features can be recovered as above.

The inventive methods provide improved compression ratios for DTED, images or graphs which have large flat regions where $|\nabla u|$ or any desired feature will be zero, hence easy to compress. Thus the inventive methods are highly accurate in these situations in retrieving the data, as well as the desired feature or features.

5

Pseudocode Appropriate for Method Involving Compressing Normals to Gradients

The following pseudocode relates to compressing a quantity, e.g. the norm of the gradient, from an original set of data, and recovering the original data from the compressed quantity.

10

```
// Read in the original file and store it as quantity u[i][j]

// Compute the derived quantity, e.g. norm of grad u[i][j]
```

15
```
// Store the norm as a separate quantity v[i][j]

%% Compression routines are completely up to the users' preference.
%% Possible compression techniques include JPEG-LS, JPEG, LSS (Level Set
%% Systems of Los Angeles, California).
```
20
```
%% LSS' technique can used with lossy compression techniques, wherein
%% there are errors between the original and restored image.


    // Compress the original file
    // Compress the normal vectors file
```
25
```
    // Decompress the compressed original file
    // Decompress the compressed normal vectors file
```

30
```
// Read in the compressed/decompressed original file — store it as
quantity uc[i][j]
```

```
// Read in the compressed/decompressed normal vectors file — store it
as quantity vc[i][j]
```

```
// Reconstruct an approximation to the original image u[i]][j], from
the compressed/decompressed quantity vc[i][j].  This reconstruction is
based upon numerically solving the partial differential equation:
```

5

```
| grad w|  = vc[i][j]        for w[i][j]
```

10
```
// As an initial guess, you may used w = uc[i][j], or set w = stored
values of u[i][j] at isolated extrema.
```

From this procedure, it can result that w[i][j] will be a better approximation to

15   u[i][j] than uc[i][j], because the relevant quantity v[i][j] will be better approximated by

w[i][j] than it will be by uc[i][j].

The Level Set Systems approach mentioned above can be found in applicant's

copending patent application, "Method and Apparatus for Feature-Based Quantization

and Compression of Data", Serial No. 09/737,834 filed December 14, 2000, which is

20   incorporated herein by reference.

Examples of Application of Methods According to the Invention

Examples 1 and 2 below will be discussed in connection with Figures 2-6.

Figure 2 illustrates a one-dimensional signal that may be divided into N pixels in

25   this manner, where N = 10 for the sake of the example. Thus, in Figures 2-6, the x-

coordinates indicating the N pixels are labeled j=1, 2, ..., 10, and the signal may be

expressed as:

$$u = \{u_j\}_{j=1}^{N}.$$

In general in the present description, an original signal will be denoted as u and its components as u_j. A reconstructed signal (i.e. data that has been regenerated from compressed data) will be denoted as v (and its components as v_j) or w (and its components as w_j), as will be seen below. In addition, a compressed signal or data will be denoted by a double bar over the compressed quantity; so $\overline{\overline{u}}$ would indicate the compressed form of the data or signal u, and similarly with any other quantities or expressions.

This simple signal is sufficient to illustrate the operation of the present invention, and it is straightforward to generalize the procedures of the invention to signals of two or more dimensions, and to real-world settings of three-dimensional phenomena.

The quantity $|\nabla u|$ (rather than u) is compressed according to the present invention, and can be computed by standard finite difference methods. Thus,

$$\left|\nabla u\right|_j = \frac{\left|u_j - u_{j-1}\right|}{\left(\frac{1}{N}\right)} = \left|u_j - u_{j-1}\right| \times N \qquad \text{for } j \geq 2; \text{ and}$$

$$\left|\nabla u\right|_1 = \frac{\left|u_2 - u_1\right|}{\left(\frac{1}{N}\right)} = \left|u_2 - u_1\right| \times N.$$

For simplicity, in these examples, we may take $u_j \geq u_{j-1}$ for all j, so we can remove the absolute value notation shown above.

Example 1

Figure 2 shows an example of a one-dimensional linear signal u, expressible as vector of data $u = \{u_j\}$, where:

$$u_j = j/N \qquad (j = 1, 2, ..., N).$$

Then $\left|\nabla u\right|_j = 1$ for all j, as shown in Figure 3. Note that $|\nabla u|$ is a constant, independent of j. Thus, the compressed value of $|\nabla u|$ is the same as the uncompressed value, as in

Figure 3. In this example, then, there is no error in the compressed and uncompressed values of $|\nabla u|$, nor is there any error in u or the reconstructed v.

In general, in this description the variable u will be used to refer to the original signal or data, and the variables v and w will be used to indicate the data after

5   reconstruction (after of the first or second gradient or derivative, as will be discussed below).

Generally, lossy compression procedures will result in errors in signals u that do not comprise constant values, so earlier methods (which compress the signals u, and not the gradient of u as in the present invention) will lead to larger errors in the gradient of the

10   reconstructed signal, i.e. $|\nabla u|$, than in the present system. As mentioned above, the methods of the invention involving compression of gradient values instead of the original signal data are applicable in one dimension or in multiple dimensions.

15   <u>Example 2</u>

Figure 4 shows an example one-dimensional signal, expressible as:

$$u_j = \frac{j(j+1)}{2N^2} \qquad (j=1, 2, ..., N)$$

Thus, for this signal $|\nabla u|_j = \frac{j}{N}$, as shown in Figure 5. (Note that the values of $|\nabla u|_j$ in Example 2 are the same as the values of u_j in Example 1, which follows since the above

20   function for u_j in Example 2 was obtained by integrating the value $u_j = j/N$ from Example 1.)

If the approach of compressing the derivative (or gradient) of the function is used, rather than compressing the function itself, then we compute $|\nabla|\nabla u||$, and obtain:

$$|\nabla|\nabla u||_j = 1 \qquad \text{for all j (as shown in Figure 6).}$$

The compression of this is error-free (for this example), and hence the errors in the reconstructed w (which approximates $|\nabla u|$) and v (which approximates u) are also zero, as is the error in the curvature approximating the curvature of $z = u(x)$. Here,

$$k = \frac{u_{xx}}{(1 + u_x^2)^{3/2}},$$

5   since there is no dependence on y. Thus, the method of Figure 7 reduces the resulting error in this example.

Using the Laplacian as described, with $\mathcal{L}(u) = u_{xx}$ (in this one-dimensional example), also results in error-free compressions of $|\nabla|\nabla u||$, $|\nabla u|$ and u.

10   In apparatus implementations of the present invention, each of the steps or operations involved can be carried out by executing one or more program or logic modules. The terrain, signal or surface data can, as discussed above, be simulated or real-world (e.g. empirically gathered) data.

A processor-based system according to the invention can thus be connected to

15   input or receiving devices (including sensors, satellite receivers, etc.) 210 and I/O channels 200 as shown in Figure 1, and the I/O channels may communicate with other apparatus that use the processed surface and gradient signals, such as avionics (not separately shown) or other user interface equipment.